

II. CLAIMS

1. (Previously Presented) A method for dynamically binding a program module in a terminal, in which one or several programs are running, said method comprising:

storing subroutines in said program module,

providing the program module with a subroutine name and parameters tag,

starting the binding by the program making a call to a subroutine,

supplementing the call with subroutine name and parameters data to select the program module for binding, in which the called subroutine is stored,

supplementing the subroutine name and parameters tag with an auxiliary tag,

supplementing the call with a public key,

comparing said subroutine name and parameters tag stored in the program module with the subroutine name and parameters data transmitted in the call,

comparing the auxiliary tag with the public key transmitted in the call, and

selecting the program module to be bound to be the program module which matches with the subroutine name and parameters data and the public key transmitted in the call.

2. (Previously Presented) The method according to claim 1, wherein the auxiliary tag to be formed in the program module contains a digital signature.

3. (Previously Presented) The method according to claim 2, further comprising verifying the digital signature of the public key formed in the program module.

4. (Previously Presented) The method according to claim 1, further comprising storing the auxiliary tag to be formed in the program module in an encrypted form.

5. (Cancelled)

6. (Previously Presented) The method according to claim 1, further comprising storing the program module in a server communicating with a digital network, wherein the terminal used is a mobile terminal, and performing the binding of the program module at least partly by messages complying with the WAP protocol.

7. (Previously Presented) A terminal comprising:

means for dynamically binding a program module, in which program module contains at least one stored subroutine and a subroutine name and parameters tag,

means for running programs,

means for starting binding by performing in the program a call to a subroutine, the call being supplemented with subroutine name and parameters data to select that program module for binding in which the called subroutine is stored, wherein the program module contains a stored auxiliary tag;

means for adding a public key to the call,

means for comparing said subroutine name and parameters tag stored in the program module with the subroutine name and parameters data transmitted in the call,

means for comparing the auxiliary tag with the public key transmitted in the call, and

means for selecting a program module to be bound on the basis of said comparison,

wherein the program module to be bound to be the program module which matches with the subroutine name and parameters data and the public key transmitted in the call.

8. (Previously Presented) The terminal according to claim 7, wherein the auxiliary tag formed in the program module contains a digital signature.

9. (Previously Presented) The terminal according to claim 8, wherein on the basis of said public key the digital signature of the auxiliary tag formed in the program module is verified.

10. (Previously Presented) The terminal according to claim 7, further comprising means for binding a program module stored in a server communicating with the Internet network, wherein the terminal is a mobile terminal, and said mobile terminal comprises means for performing binding of the program module at least partly by messages complying with the WAP protocol.

11. (Previously Presented) A method in a terminal, in which one or several programs are running, said terminal comprising at least one program module, which is provided with a subroutine name and parameters tag and an auxiliary tag, and said at least one program module comprises at least one subroutine, wherein the method comprises:
making a call by the program to a subroutine;

supplementing the call with first call data to select a program module for dynamically binding among said at least one program module, in which the called subroutine is stored;

supplementing the call with a public key in connection with the binding;

comparing said subroutine name and parameters tag stored in said at least one program module with the subroutine name and parameters data transmitted in the call;

comparing the auxiliary tag with the public key transmitted in the call; and

selecting the program module for binding to be the program module in which the subroutine name and parameters tag matches with the subroutine name and parameters data and the auxiliary tag matches with the public key.

12. (Previously Presented) A terminal comprising:

a binding server for dynamically binding a program module, which program module contains at least one subroutine, a subroutine name and parameters tag and an auxiliary tag;

an element for running programs and configured to:

perform in the program a call to a subroutine,

supplementing the call with subroutine name and parameters data to select that program module for binding in which the called subroutine is stored;

supplementing the call with a public key;

comparing said subroutine name and parameters tag stored in the program module with the subroutine name and parameters data transmitted in the call;

comparing the auxiliary tag with the public key transmitted in the call; and

selecting a program module to be bound on the basis of said comparison,

Serial No. 09/739,941

Response to the Advisory Office Action dated 2 April 2007

wherein the program module to be bound to be the program module which matches with the subroutine name and parameters data and the public key transmitted in the call.